

	Type	L #	Hits	Search Text	DBs	Time Stamp
1	BRS	L1	1	09/921,841	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:09
2	BRS	L2	1	(wave or carrier or transm\$5) and 1	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:14
3	BRS	L3	3723	(713/1 or 713/2 or 713/100).ccls.	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:14
4	BRS	L4	11	(modif\$4 or chang\$4 or alter\$4 or upgrad\$4 or updat\$4) with (firmware or software or program\$4 or instruction or code) with (boot\$4 or reboot\$4) with (cache or private or local) with memory	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:18

	Type	L #	Hits	Search Text	DBs	Time Stamp
5	BRS	L5	4	4 and 3	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:17
6	BRS	L6	10	((modif\$4 or chang\$4 or alter\$4 or upgrad\$4 or updat\$4) same (firmware or software or program\$4 or instruction or code) same (boot\$4 or reboot\$4) same (cache or private or local) same memory).clm.	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:23
7	BRS	L7	4	6 and 4	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:19
8	BRS	L8	3	7 not 5	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:19

	Type	L #	Hits	Search Text	DBs	Time Stamp
9	BRS	L9	22	((modif\$4 or chang\$4 or alter\$4 or upgrad\$4 or updat\$4) same (firmware or software or program\$4 or instruction or code) same ("not" or without or no) adj5 (boot\$4 or reboot\$4) same (cache or private or local) same memory)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:24
10	BRS	L10	4	6 and 3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:23
11	BRS	L12	0	((modif\$4 or chang\$4 or alter\$4 or upgrad\$4 or updat\$4) same (firmware or software or program\$4 or instruction or code) same ("not" or without or no) adj5 (boot\$4 or reboot\$4) same (cache or private or local) same memory)).clm.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:24

	Type	L #	Hits	Search Text	DBs	Time Stamp
12	BRS	L11	10	9 and 3	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:32
13	BRS	L13	1	"6272629".pn.	US-PGPUB ; USPAT; EPO; JPO; IBM_TDB	2006/06/05 08:33

US-PAT-NO: 5835761

DOCUMENT-IDENTIFIER: US 5835761 A

TITLE: Information processing system capable of updating a BIOS programme without interrupting or stopping the operational of a system

DATE-ISSUED: November 10, 1998

US-CL-CURRENT: 713/100, 713/2

APPL-NO: 08/895529

DATE FILED: July 17, 1997

PARENT-CASE:

This application is a continuation, of application Ser. No. 08/363,690, filed Dec. 23, 1994, now abandoned.

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
GB	9413088	June 29, 1994

----- KWIC -----

Brief Summary Text - BSTX (16):

In other variations of the above-described system, the update programme input means may be a floppy diskdrive, an IC card, a serial port connection, or a local area network connection. In each of these variations, the operating system provided may be of a multitasking type. When a multitasking operating system is used, an updated basic input/output system programme may be loaded into memory while a current basic input/output system programme is being

executed out of memory. A software switch may then be automatically operated to cause the operating system to begin executing the updated basic input/output system programme, without the need to shut down and reboot the system.

**Current US Original Classification - CCOR (1):
713/100**

**Current US Cross Reference Classification - CCXR
(1):
713/2**

US-PAT-NO: 6032239

DOCUMENT-IDENTIFIER: US 6032239 A

TITLE: System and method for updating partition mappings to logical drives in a computer memory device

DATE-ISSUED: February 29, 2000

US-CL-CURRENT: 711/173, 713/100

APPL-NO: 08/950545

DATE FILED: October 15, 1997

----- KWIC -----

Abstract Text - ABTX (1):

A system and method for changing partition mappings to logical drives in a memory device of a computer. The computer includes a processor for running operating system code and a hard drive. The hard drive, which can be accessed by the processor, includes at least one partition. When the computer boots up, a logical reference value in the operating system code points to the partition. After the partition has been initialized or modified, a reference-updating routine is run for dynamically updating the logical reference value to correctly map to the partition. The reference-updating routine first locates and flushes any caches in the computer and closes any open files accessible by the operating system code. The routine then loads a boot record for the partition and retrieves specific data therefrom. With the retrieved data, the routine updates the logical reference value. Therefore, after the

reference-updating routine has performed the updating, the processor may continue running the operating code without requiring the computer to reboot.

Current US Cross Reference Classification - CCXR

(1):

713/100

US-PAT-NO: 6754828

DOCUMENT-IDENTIFIER: US 6754828 B1

TITLE: Algorithm for non-volatile memory updates

DATE-ISSUED: June 22, 2004

US-CL-CURRENT: 726/2, 710/22 , 713/1 , 713/187 , 713/2 , 719/310 , 726/16

APPL-NO: 09/352715

DATE FILED: July 13, 1999

----- KWIC -----

Detailed Description Text - DETX (17):

In certain embodiments of non-volatile memory such as FLASH memory, further writes to the non-volatile memory blocks containing code will be prevented by setting certain lock bits associated with the non-volatile memory. After successfully completing the update of the relevant code blocks in the non-volatile memory, in one embodiment, the SAL procedure can reset the electronic system, so that all the processors will load and execute the new input binary, or updated firmware. In an alternative embodiment, the OS will copy the new input binary into other memory, e.g. RAM, of the electronic device and bring all of the other processors to an active state with SAL coordination

460. The OS and/or SAL firmware will also take appropriate actions to eliminate portions of the old binary that may be present within the processors' cache memory. If the system is to execute the updated firmware without rebooting the system, the Operating system has to co-operate with the update

process and re-register all the PAL firmware procedural entry points with the SAL firmware and provide appropriate mappings for the same. In this embodiment, there is no down time for the electronic system and the electronic system functions as a highly reliable, available and serviceable (RAS) system.

Current US Cross Reference Classification - CCXR

(2):

713/1

Current US Cross Reference Classification - CCXR

(4):

713/2

PGPUB-DOCUMENT-NUMBER: 20020188837

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20020188837 A1

TITLE: Booting to a recovery/maintenance environment

PUBLICATION-DATE: December 12, 2002

US-CL-CURRENT: 713/2

APPL-NO: 09/876426

DATE FILED: June 7, 2001

----- KWIC -----

**Current US Classification, US Primary
Class/Subclass - CCPR (1):
713/2**

Summary of Invention Paragraph - BSTX (12):
[0009] In one embodiment of the present invention, a method for booting to a partition without a local operator using a BOOT register may comprise the step of an operating system determining that one or more programs within the partition in a nonvolatile storage unit may need to be accessed to resolve a problem, e.g., corrupt driver, hardware failure, operating system program in main memory became damaged. Once the operating system determines that the partition needs to be accessed, the operating system may search a table such as a Root System Description Table (RSDT) for a pointer to the location of the BOOT register. The BOOT register may comprise a plurality of bits where one or

more bits may be used to communicate to the BIOS as to what actions need to be taken on the next boot. The operating system may set one or more bits in the BOOT register to indicate to the BIOS as to whether or not to boot to the partition to conduct recovery/maintenance operations on the subsequent boot. The one or more bits set by the operating system may indicate to the BIOS to perform one or more activities, e.g., loading the original copy of the operating system, executing code to run a diagnosis, enabling remote problem determination and correction, updating routines, on the subsequent boot to the partition. Operating system may then issue a request to the BIOS to shut down or restart the processing unit which allows the BIOS to invoke a series of system checks commonly referred to as Power-On Self-Tests (POST). During the POST, the BIOS may then read the BOOT register and determine whether or not to boot to the partition as well as any activities to perform if the BIOS is to boot to the partition.

Detail Description Paragraph - DETX (4):

[0019] FIG. 1 illustrates a typical hardware configuration of data processing system 113 which is representative of a hardware environment for practicing the present invention. Data processing system 113 has a central processing unit (CPU) 110 coupled to various other components by system bus 112. An operating system 150 runs on CPU 110 and provides control and coordinates the function of the various components of FIG. 1. An application 170 runs in conjunction with operating system 150 and provides output calls to operating system 150 which implements the various functions or services to be performed by application 170. Read only memory (ROM) 116 is coupled to

system

bus 112 and includes a basic input/output system ("BIOS") that controls certain

basic functions of data processing system 113. Nonvolatile memory 165 is coupled to system bus 112 and may comprise a BOOT register 160 thereby providing a mechanism for operating system 150 to communicate to the BIOS about

what actions need to be taken on the next boot. A more detailed description of

operating system 150 writing to BOOT register 160 and the BIOS reading BOOT

register 160 during POST to determine what actions to take on a subsequent boot

is provided in the description of FIG. 2. Random access memory (RAM) 114, I/O

adapter 118, and communications adapter 134 are also coupled to system bus 112.

It should be noted that software components including operating system 150 and

application 170 are loaded into RAM 114 which is the computer system's main

memory. I/O adapter 118 may be an integrated drive electronics ("IDE") adapter

that communicates with disk unit 120, e.g., disk drive. Disk unit 120 may comprise a "hidden" partition 190 invisible to operating system 150 and resistant to modification by any programs running under operating system 150

including operating system 150 itself. The hidden partition 190 may store an

original copy of the operating system files and the code to restore them in case of accidental corruption of the operating system program loaded in main

memory 114. The hidden partition 190 may further store code to invoke diagnostic routines to inspect suspected hardware failures. Furthermore, the

hidden partition 190 may be used to store code that enables remote diagnosis in

the case of a system crash. The hidden partition 190 may further be used to

store system code update routines, e.g., BIOS flash update. It is noted that

the "hidden" partition 190 may reside in any non-volatile storage device and that FIG. 1 is illustrative. It is further noted that the program of the present invention that allows the BIOS to boot to a designated partition in a non-volatile storage unit, e.g., disk unit 120, to conduct recovery/maintenance operations without requiring a local operator using BOOT register 160, as described in FIG. 2, may reside in disk unit 120 or in application 170.

US-PAT-NO: 6807643

DOCUMENT-IDENTIFIER: US 6807643 B2

TITLE: Method and apparatus for providing diagnosis of a processor without an operating system boot

DATE-ISSUED: October 19, 2004

US-CL-CURRENT: 714/36, 713/2

APPL-NO: 09/469249

DATE FILED: December 22, 1999

PARENT-CASE:

CLAIM FOR PRIORITY

This is a continuation-in-part (CIP) application from an application for "Method And Apparatus For Establishing Network Connection For A Processor Without An Operating System Boot" filed in the United States Patent & Trademark Office on Dec. 29, 1998, assigned Ser. No. 09/221,575, now issued as U.S. Pat. No. 6,272,629.

----- KWIC -----

Brief Summary Text - BSTX (8):

U.S. Ser. No. 09/221,575 filed by David C. Stewart, one of the co-inventors of the present application, on Dec. 29, 1998 discloses a method and apparatus which permits establishing a network connection for a processor without an operating system boot. In this arrangement, pre-boot services are

loaded into a volatile memory of the processor from a nonvolatile memory. These pre-boot services include code for a modem driver. The modem driver code

is then used to establish the network connection. This permits remote diagnosis and repair (as well as updating) even when the processor will not boot, not only in Local-Area-Networks (LANs) such as the Intel Wired-for-Manageability (WfM), but also in the so-called "occasionally connected" computers. As such, the arrangement disclosed in Ser. No. 09/221,575 provides a good starting point for developing methods and systems

that are capable of both self-diagnosis and remote repair functions. The present invention is directed to further developments of the disclosure set forth in Ser. No. 09/221,575.

Current US Cross Reference Classification - CCXR

(1):

713/2